Our Docket No.: 00-251 / 1496.00049

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Applicant:	Zhaohui Shen et al.		
Application No.:	09/684,868	Examiner:	Dinh, P.
Filed:	October 6, 2000	Art Group:	2825
For:	DIAGNOSTIC ARCHITECTURE USING FPGA CORE IN SYSTEM ON A CHIP DESIGN		

CERTIFICATE OF MAILING

I hereby certify that this letter, the response or amendment attached hereto are being deposited with the United States Postal Service as first class mail in an envelope addressed to Mail Stop Appeal Brief Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on June 3, 2004.

By: Jan M Dunbar
Jan M. Dunbar

APPEAL BRIEF

Mail Stop - Appeal Brief Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Appellants submit, in triplicate, the following Appeal Brief pursuant to 37 C.F.R. §1.192 for consideration by the Board of Patent Appeals and Interferences. Please charge $330.00 to cover the cost of filing the opening brief as required by 37 C.F.R. §1.17(c) and any additional fees or credit any overpayment to Deposit Account Number 12-2252.

IFW AF/2855

Attorney Docket: 00-255/1496.00039

IN RE APPLICATION OF: Zhaohui Shen et al.

SERIAL NO.: 09/684,868

RESPONSE TRANSMITTAL AND EXTENSION OF TIME REQUEST (IF REQUIRED)

TITLE: DIAGNOSTIC ARCHITECTURE USING FPGA CORE IN SYSTEM ON A CHIP DESIGN

FILED: October 6, 2000

EXAMINER: Dinh, P.

ART UNIT: 2825

COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Enclosed please find an amendment and a postcard along with the fee calculation below:

FEE CALCULATION FOR ENCLOSED AND EXTENSION REQUEST (IF ANY)

	Claims Remaining	Highest No. Previous	Extra Rate	Additional Fee
Total Claims	30 minus	30 =	0 x $ 18.00	$0.00
Independent Claims	2 minus	3 =	0 x $ 84.00	$ 0.00

Multiple Dependent Claim First Added + $280.00 $ 0.00

TOTAL IF NOT SMALL ENTITY .. $0.00

[ ] SMALL ENTITY STATUS - If applicable, divide by 2 .... $0.00
[ ] Verified statement enclosed, if not previously filed.

[ ] Applicant also requests a ____ month extension of time for response to the outstanding Office Action. The fee is $0.00

[X] Fee set forth for filing Appeal Brief $330.00

TOTAL FEE $330.00

The Commissioner is hereby authorized to charge all fees associated with this Communication to Deposit Account No. 12-2252. A duplicate copy of this sheet is attached.

CHRISTOPHER P. MAIORANA, P.C.

24840 Harper Avenue, Suite 100
St. Clair Shores, Michigan 48080
(586) 498-0670

By: ______________________
Christopher P. Maiorana
Registration No.: 42,829

I hereby certify that this letter, the response or amendment attached hereto are being deposited with the United States Postal Service as first class mail in an envelope addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on June 3, 2004.

By: Jan M. Dunbar
Jan M. Dunbar

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

The real party in interest is the Assignee, LSI Logic Corporation.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences known to the Appellants, Appellants' legal representative, or Assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1, 4-9 and 11-33 are pending and remain rejected. The Appellants hereby appeal the rejection of claims 1, 4-9 and 11-33.

IV. STATUS OF AMENDMENTS

Appellants are appealing a Final Office Action issued by the Examiner on November 4, 2003. On February 4, 2004, Appellants filed an Amendment After Final that was not entered by the Examiner. On April 5, 2004, Appellants filed a Notice of Appeal based on the last set of claims prior to the Amendment After Final.

V. SUMMARY OF INVENTION

The present invention concerns a system (FIG. 1 reference no. 100) for designing an integrated circuit (IC) (FIG. 1 block 102, page 6 lines 1-8) generally comprising a functional portion (FIG. 2 block 112), a logic portion (FIG. 2 block 116), a debugging/bug fix circuit (FIG. 1 block

104) and a diagnostic architecture using an FPGA core in a system on a chip design (FIG. 2 blocks 110, 114 and 116, page 4 lines 16-18). The logic portion may be connected to the functional portion (FIG. 2 output of MUX 134, FIG. 5 output of Module 202a and FIG. 5 input to Module 202n) and generally configured to detect errors, fix errors or verify fixes of errors in the functional portion (page 6 lines 10-13, page 17 lines 17-20). The logic portion may include one or more interfaces. (FIG. 2 bus 152, page 6 lines 13-14). The debugging/bug fix circuit may be configured to detect errors in the logic portion through the one or more interfaces (page 6 lines 15-16).

VI. ISSUES

The issue is whether claims 1, 4-9 and 11-33 are patentable under 35 U.S.C. §102(e) over Killian et al., U.S. Patent No. 6,477,683 (hereafter Killian).

VII. GROUPING OF CLAIMS

Appellants contend that the claims of the present invention do not stand or fall together. In particular, the following groups of claims are separately patentable:

Group 1:	Claims 1, 4, 5, 6, 8, 9, 11, and 13-25 stand together.
Group 2:	Claims 26, 28, 29 and 33 stand together.
Group 3:	Claim 7 stands alone.
Group 4:	Claim 12 stands alone.
Group 5:	Claim 27 stands alone.
Group 6:	Claim 30 stands alone.
Group 7:	Claim 31 stands alone.

Group 8: Claim 32 stands alone.

The claim(s) in each group is (are) separately patentable from the claim(s) in any other groups.

VIII. ARGUMENTS

A. 35 U.S.C. § 102

The Federal Circuit has stated that "[t]o anticipate, *every element and limitation* of the claimed invention must be found in a single prior art reference, *arranged as in the claim.*"[1] (Emphasis added). The Federal circuit has added that the anticipation determination is viewed from one of ordinary skill in the art: "There must be no difference between the claimed invention and the reference disclosure, as viewed by a person of ordinary skill in the field of the invention."[2] Furthermore, "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference."[3]

[1] *Brown v. 3M,* 256 F.3d 1349, 1351, 60 USPQ2d 1375, 1376 (Fed. Cir. 2001) citing *Karsten Mfg. Corp. v. Cleveland Golf Co.*, 242 F.3d 1376, 1383, 58 USPQ2d 1286, 1291 (Fed. Cir. 2001); *Scripps Clinic & Research Found. v. Genentech Inc.*, 927 F.2d 1565, 18 U.S.P.Q.2d 1001, 1010 (Fed. Cir. 1991) (Emphasis added).

[2] *Scripps Clinic & Research Found. v. Genentech Inc.*, 927 F.2d 1565, 18 U.S.P.Q.2d 1001, 1010 (Fed. Cir. 1991).

[3] *Verdegaal Bros. V. Union Oil Co. of California*, 814 F.2d 628, USPQ2d 1051, 1053 (Fed Circ. 1987).

1. Group 1 (claims 1, 4, 5, 6, 8, 9, 11, and 13-25) is fully patentable over Killian

The claims of group 1 provide a functional portion and a logic portion connected to the functional portion. In contrast, the Examiner does not provide evidence of elements expressly found in Killian similar to the claimed elements as required by *Verdegaal Bros*. In particular, the Examiner does not identify any elements in Killian allegedly similar to the claimed functional portion or the claimed logic portion. The Examiner also does not provide any evidence that Killian explicitly discloses the claimed connection as required by *Brown*. The only evidence provided by the Examiner is "fig 1-3, 5-15"[4], which appears to be an improper omnibus argument per M.P.E.P. §707.07(d). Therefore, *prima facie* anticipation is not established for a functional portion and a logic portion connected to the functional portion as presently claimed.

The claims of group 1 further provide that the logic portion includes one or more interfaces. In contrast, the Examiner does not provide evidence of similar interfaces expressly found in Killian. In particular, the Examiner merely provides a substantial list of interfaces mentioned in Killian.[5] No evidence exists that any of the listed interfaces are arranged as in the claimed logic portion. Therefore, *prima facie* anticipation is not established for a logic portion including one or more interfaces as presently claimed.

The claims of group 1 further provide a debugging/bug fix circuit configured to detect errors in the logic portion through the one or more interfaces. In contrast, the Examiner does not provide evidence that an on-chip debug module 92 found in Killian (asserted similar to the claimed debugging/bug fix circuit) can (i) detect errors in some unidentified element allegedly similar to the

[4] Office Action, November 4, 2003, page 3, Claim 1 arguments.

[5] Office Action, November 4, 2003, page 3, Claim 1 arguments.

claimed logic portion (ii) through some unidentified interfaces allegedly similar to the claimed one or more interfaces. Furthermore, the Examiner fails to provide evidence for a structure connecting the on-chip debug module 92 of Killian to the unidentified element allegedly similar to the claimed logic portion. Therefore, *prima facie* anticipation is not established for a debugging/bug fix circuit configured to detect errors in a logic portion through one or more interfaces as presently claimed.

The claims of group 1 further provide a diagnostic architecture using an FPGA core in a system on a chip design. In contrast, the Examiner fails to provide evidence of a diagnostic architecture, an FPGA core or a system on a chip design. In particular, no evidence is presented that an on-chip debug module 92 of Killian uses an FPGA core. Furthermore, the text of Killian cited by the Examiner, reproduced below, appears to be silent regarding the claim limitation:[6]

> Unlike design of the ISA and processor, system design (which may include the design of chips that now include the processor) is quite common and systems are typically continuously designed.[7]
>
> Also, based on the system architecture and specifications of chip foundries, a chip foundry is chosen based on an evaluation of foundry capabilities with respect to the system HDL (not related to processor selection as in the prior art).[8]
>
> On-chip debug module: yes, no
> Full scan: yes. no
> Interrupts
> Source: external, software
> Priority level
> System Memory Addresses
> Vector and address calculation method: XTOS, manual
> Configuration Parameters
> RAM size, start address: arbitrary

[6] Office Action, November 4, 2003, page 3, last line.

[7] Killian, column 2, lines 2-3.

[8] Killian, column 7, lines 32-33.

> ROM size, start address: arbitrary
> XTOS: arbitrary
> Configuration Specific Addresses
> User exception vector: arbitrary
> Kernel Exception vector: arbitrary
> Register window over/underflow vector base: arbitrary
> Reset vector: arbitrary
> XTOS start address: arbitrary
> Application start address: arbitrary
> TIE Instructions
> (define ISA extensions)
> Target CAD Environment
> Simulation
> Verilog.TM.: yes, no
> Synthesis
> Design Compiler.TM.: yes, no
> Place & Route
> Apollo.TM.: yes, no
> Additionally, the system 10 may provide options for adding other functional units such as a 32-bit integer multiply/divide unit or a floating point arithmetic unit; a memory management unit; on-chip RAM and ROM options; cache associativity; enhanced DSP and coprocessor instruction set; a write-back cache; multiprocessor synchronization; compiler-directed speculation; and support for additional CAD packages.[9]
>
> The system of claim 63, wherein the performance characteristics include at least one of area required to implement the processor on a chip, power consumed by the processor and clock speed of the processor.[10]

Nowhere in the above text does Killian expressly discuss (i) a diagnostic architecture, (ii) an FPGA core or (iii) a system on a chip design. Therefore, *prima facie* anticipation is not established for a diagnostic architecture using an FPGA core in a system on a chip design as presently claimed.

In summary, *prima facie* anticipation is not established for the claims of group 1 due to a lack of evidence that Killian expressly discloses all of the claim limitations. *Prima facie* anticipation is not established for lack of evidence that the relied-upon elements of Killian are arranged as in the

[9] Killian, column 12, lines 3-40.

[10] Killian, column 48, claim 65.

claims. Furthermore, several of the claimed elements are rejected based upon improper omnibus arguments that fail to identify specific elements and connections between the elements in Killian. As such, group 1 is fully patentable over the cited reference and the rejection should be reversed.

2. Group 2 (claims 26, 28, 29 and 33) is fully patentable over Killian

The claims of group 2 provide a step for interfacing a chip with an FPGA core. In contrast, the Examiner does not provide evidence of a step expressly found in Killian similar to the claimed step as required by *Verdegaal Bros.* In particular, the text of Killian cited by the Examiner, reproduced below, appears to be silent regarding an interfacing step:[11]

> One such example is FPGA Express from Synopsys.[12]
>
> A fully customized implementation script to be used by the vendor tools is also generated.[13]

Nowhere in the above text does Killian expressly discuss interfacing a chip with an FPGA core. Furthermore, the assertion that the claimed step may be found in FIGS. 1-15 of Killian is an improper omnibus argument per M.P.E.P. §707.07(d).[14] Therefore, *prima facie* anticipation is not established for a step of interfacing a chip with an FPGA core as presently claimed.

The group 2 claims further provide a step for presenting one or more internal signals of the chip. In contrast, the Examiner does not provide any evidence that a similar presenting step is expressly found in Killian. Furthermore, the cite by the Examiner to all of the figures of Killian is

[11] Office Action, November 4, 2003, page 5, claim 26 arguments.

[12] Killian, column 32, lines 54-55.

[13] Killian, column 33, lines 2-4.

[14] Office Action, November 4, 2003, page 5, Claim 26 arguments, line (A).

an improper omnibus argument per M.P.E.P. §707.07(d). Therefore, *prima facie* anticipation is not established for a step of presenting one or more internal signals of a chip as presently claimed.

The group 2 claims further provide a step for verifying or fixing bugs in the chip with the one or more internal signals. In contrast, the Examiner does not provide evidence that Killian expressly discloses a verifying or fixing step. Furthermore, the cite by the Examiner to all of the figures, the entire abstract and the entire background of Killian is an improper omnibus argument per M.P.E.P. §707.07(d). Therefore, *prima facie* anticipation is not established for a step of verifying or fixing bugs in a chip with one or more internal signals as presently claimed.

The group 2 claims further provide a step for programming an FPGA core to dump data from a host register every N clock cycles. In contrast, the Examiner provides a long list of cites allegedly disclosing the claimed step. However, Killian only expressly mentions an "FPGA" element once as follows:

> Since one of the purposes of the emulation board is to support quick prototype implementation for debugging purposes, it is important that the CPLD implementation process outlined in the previous paragraph is automatic. To achieve this objective, the files delivered to users are customized by grouping all relevant files into a single directory. Then, a fully customized synthesis script is provided to be able to synthesize the particular processor configuration to the particular **FPGA device** selected by the customer.[15] (Emphasis added.)

Nowhere in the above text does Killian discuss a step for programming the FPGA to dump data from a host register. Therefore, *prima facie* anticipation is not established for a step of programming an FPGA core to dump data from a host register every N clock cycles as presently claimed.

In summary, *prima facie* anticipation is not established for lack of express evidence that Killian discloses all of the claimed steps. Furthermore, several of the claimed elements are rejected

[15] Killian, column 32, line61 thru column 33, line 2.

based upon improper omnibus arguments that fails to identify specific steps in Killian. As such, group 2 is fully patentable over the cited reference and the rejection should be reversed.

3. Group 3 (claim 7) is fully patentable over Killian

The group 3 claim contains all of the limitations of group 1. Therefore, the arguments presented above in support of the patentability of group 1 are incorporated hereunder in support of group 3.

The group 3 claim further provides that the system is configured to allow the debugging/bug fix circuit observation of one or more signals when running in a normal mode. In contrast, the text of Killian cited by the Examiner does not expressly discuss the on-chip debugging circuit 92 (asserted similar to the claimed debugging/bug fix circuit) to observe one or more signals in a normal mode.[16] The text of Killian cited by the Examiner reads:

> The module 92 provides support for exception generation to put the processor 60 **in the debug mode;** access to all program-visible registers or memory locations; execution of any instruction that the processor 60 is configured to execute; modification of the PC to jump to a desired location in the code; and a utility to allow **return to a normal operation mode**, triggered from outside the processor 60 via the JTAG port 94. (Emphasis added) [17]

The above text appears to contemplate operating in a debug mode with a capability to return to a normal mode. Nothing is said how the module 92 operates with the processor 60 in the normal mode. Therefore, Killian does not disclose or suggest allowing a debugging/bug fix circuit observation of one or more signals when running in a normal mode as presently claimed. As such, group 3 is fully patentable over the cited reference and the rejection should be withdrawn.

[16] Office Action, November 4, 2003, page 4, claim 7-8 arguments.

[17] Killian, column 13, lines 16-23.

4. Group 4 (claim 12) is fully patentable over Killian

The group 4 claim contains all of the limitations of group 1. Therefore, the arguments presented above in support of the patentability of group 1 are incorporated hereunder in support of group 4.

The group 4 claim further provides that the debugging/bug fix circuit comprises a debugging workstation. In contrast, the text and FIG. 1 of Killian cited by the Examiner appear to be silent regarding a workstation that performs debugging. In particular, the blocks in FIG. 1 of Killian are labeled User Interface 20, Build System 50, Software Development Tool 30 and Hardware Description 40. None of the blocks in FIG. 1 of Killian expressly mention a debugging workstation. The text of Killian cited by the Examiner also appears to be silent regarding a debugging workstation:

> The configurable ISS [Instruction Set Simulator] 126 is used for the following four purposes or goals as part of the system design and verification process:
> debugging software applications before hardware becomes available;[18]
>
> In this way, all ISS configurable behavior is derived from well-defined sources related to other parts of the system.[19]
>
> The serial channels 212 provide a communication link to UNIX or PC hosts for downloading and debugging user programs.[20]

Nowhere in the above text does Killian expressly discuss a debugging workstation. Neither the Instruction Set Simulator 126 nor UNIX/PC hosts mentioned above are described as being operational to detect errors in a logic portion through one or more interfaces, as incorporated into

[18] Killian, column 30, lines 36-40.

[19] Killian, column 31, lines 11-13.

[20] Killian, column 32, lines 33-35.

group 4 from group 1. Therefore, Killian does not appear to disclose or suggest a debugging/bug fix circuit comprising a debugging workstation as presently claimed. As such, group 4 is fully patentable over the cited reference and the rejection should be reversed.

5. Group 5 (claim 27) is fully patentable over Killian

The group 5 claim contains all of the limitations of group 2. Therefore, the arguments presented above in support of the patentability of group 2 are incorporated hereunder in support of group 5.

The group 5 claim further provides a computer readable medium configured to store instructions for executing the steps of (A) interfacing a chip with an FPGA core, (B) presenting one or more internal signals of the chip, (C) verifying or fixing bugs in the chip with the one or more internal signals and (D) programming the FPGA core to dump data from a host register every N clock cycles (steps A-D from claim 26). In contrast, the text of Killian cited by the Examiner and FIG. 1 of Killian appear to be silent regarding a computer readable medium storing instructions. In particular, the blocks in FIG. 1 of Killian are labeled User Interface 20, Build System 50, Software Development Tool 30 and Hardware Description 40. None of the blocks in FIG. 1 of Killian expressly mention a computer readable medium. The text of Killian cited by the Examiner reads as follows:

> Place & Route
> Apollo.TM.: yes, no[21]

[21] Killian, column 12, lines 30-31.

> The configuration process begins with software tools 30 that can be ported to a variety of different systems and instruction set architectures.[22]
>
> The ISS 126 allows programs generated for the configured processor 60 to be executed on a host computer.[23]
>
> In this way the system provides a design for flexibly moving these services between ISS host and simulation target.[24]
>
> The serial channels 212 provide a communication link to UNIX or PC hosts for downloading and debugging user programs.[25]

Nowhere in the above text does Killian expressly mention a computer readable medium configured to store instructions for executing steps. Therefore, Killian does not disclose or suggest a computer readable medium configured to store instructions for executing the steps of (A) interfacing a chip with an FPGA core, (B) presenting one or more internal signals of the chip, (C) verifying or fixing bugs in the chip with the one or more internal signals and (D) programming the FPGA core to dump data from a host register every N clock cycles as presently claimed. As such, group 5 is fully patentable over the cited reference and the rejection should be reversed.

6. Group 6 (claim 30) is fully patentable over Killian

The group 6 claim contains all of the limitations of group 2. Therefore, the arguments presented above in support of the patentability of group 2 are incorporated hereunder in support of group 6.

[22] Killian, column 25, line 67- column 26, line 2.

[23] Killian, column 30, lines 20-21.

[24] Killian, column 31, lines 11-13.

[25] Killian, column 32, lines 34-36.

The group 6 claim further provides a step for searching for a specific signal pattern. In contrast, the text of Killian cited by the Examiner does not expressly mention a step for searching for signal patterns. The text of Killian cited by the Examiner reads:

> In this system the utility function is constructed from the input goals. For example, given the goals Performance>200, Power<100, Area<4, with the priority of Power, Area, and Performance, the following utility function could be used:
>
> Max((1-Power/100)*0.5, 0)+(max((1-Area/4)*0.3, 0)*(if Power <100 then 1 else (1-Power/100)**2))+(max(Performance/200*0.2, 0)*(if Power<100 then 1 else (1-Power/100)**2))*(if Area<4 then 1 else (1-area/4)**2))
>
> which rewards decreases in power consumption until it is below 100 and then is neutral, rewards decreases in area until it is below 4, and then is neutral, and rewards increases in performance until it is above 200, and then is neutral. There are also components that reduce the area usage when power is out of spec and that reduce the performance usage when power or area are out of spec.
>
> Both these algorithms and others can be used to search for configurations that satisfy the specified goals. What is important is that the configurable processor design has been described in a design database that has prerequisite and incompatibility option specifications and the impact of the configuration options on various metrics.
>
> The examples we have given have used hardware goals that are general and not dependent on the particular algorithm being run on the processor 60. The algorithms described can also be used to select configurations well suited for specific user programs. For example, the user program can be run with a cache accurate simulator to measure the number of cache misses for different types of caches with different characteristics such as different sizes, different line sizes and different set associativities. The results of these simulations can be added to the database used by the search algorithms 106 described to help select the hardware implementation description 40.
>
> Similarly, the user algorithm can be profiled for the presence of certain instructions that can be optionally implemented in hardware. For example, if the user algorithm spends a significant time doing multiplications, the search engine 106 might automatically suggest including a hardware multiplier. Such algorithms need not be limited to considering one user algorithm. The user can feed a set of algorithms into the system, and the search engine 106 can select a configuration that is useful on average to the set of user programs.
>
> In addition to selecting preconfigured characteristics of the processors 60, the search algorithms can also be used to automatically select or suggest to the users possible TIE extensions. Given the input goals and given examples of user programs written perhaps in the C programming language, these algorithms would suggest potential TIE extensions. For TIE extensions without state, compiler-like tools can be embodied with pattern matchers. These pattern matchers walk expression nodes in a bottom up fashion searching for multiple instruction patterns that could be replaced with a single instruction. For example, say that the user C program contains the following statements.
>
> ```
> x=(y+z)<<2;
> x2=(y2+z2)<<2;
> ```
>
> The pattern matcher would discover that the user in two different locations adds two numbers and shifts the result two bits to the left. The system would add to a database the possibility of generating a TIE instruction that adds two numbers and shifts the result two bits to the left.
>
> The build system 50 keeps track of many possible TIE instructions along with a count of how many times they appear.[26]

[26] Killian, column 18, line 67-column 20, line 1.

> Other extensions include looking at a group of configuration parameters at a time or employing more sophisticated searching algorithms.[27]

Nowhere in the above text does Killian expressly disclose searching for specific patterns in signals. Furthermore, the reference to FIGS. 6-15 of Killian is an improper omnibus argument per M.P.E.P. §707.07(d).[28] Therefore, Killian does not disclose or suggest a step for searching for a specific signal pattern as presently claimed. As such, group 6 is fully patentable over the cited reference and the rejection should be reversed.

7. Group 7 (claim 31) is fully patentable over Killian

The group 7 claim contains all of the limitations of group 6. Therefore, the arguments presented above in support of the patentability of group 6 are incorporated hereunder in support of group 7.

The group 7 claim further provides a step for monitoring a correctness of a bus protocol. In contrast, the text of Killian cited by the Examiner appears to be silent regarding monitoring bus protocols. The text of Killian cited by the Examiner reads:

> As an example, consider a device designed to transmit and receive data over a channel using a complex protocol. Because the protocol is complex, the processing cannot be reasonably accomplished entirely in hard-wired, e.g., combinatorial, logic, and instead a programmable processor is introduced into the system for protocol processing. Programmability also allows bug fixes and later upgrades to protocols to be done by loading the instruction memories with new software. However, the traditional processor was probably not designed for this particular application (the application may not have even existed when the processor was designed), and there may be operations that it needs to perform that require many instructions to accomplish which could be done with one or a few instructions with additional processor logic.

[27] Killian, column 35, lines 41-43.

[28] Office Action, November 4, 2003, page 5, Claim 30 arguments.

> Because the processor cannot easily be enhanced, many system designers do not attempt to do so, and instead choose to execute an inefficient pure-software solution on an available general-purpose processor. The inefficiency results in a solution that may be slower, or require more power, or be costlier (e.g., it may require a larger, more powerful processor to execute the program at sufficient speed). Other designers choose to provide some of the processing requirements in special-purpose hardware that they design for the application, such as a coprocessor, and then have the programmer code up access to the special-purpose hardware at various points in the program. However, the time to transfer data between the processor and such special-purpose hardware limits the utility of this approach to system optimization because only fairly large units of work can be sped up enough so that the time saved by using the special-purpose hardware is greater than the additional time required to transfer data to and from the specialized hardware.
>
> In the communication channel application example, the protocol might require encryption, error-correction, or compression/decompression processing.[29]

Nowhere in the above text does Killian expressly mention monitoring bus protocols for correctness. Furthermore, the reference to all of the figures of Killian by the Examiner is an improper omnibus argument per M.P.E.P. §707.07(d).[30] Therefore, Killian does not disclose or suggest a step for monitoring a correctness of a bus protocol as presently claimed. As such, group 7 is fully patentable over the cited reference and the rejection should be reversed.

8. Group 8 (claim 32) is fully patentable over Killian

The group 8 claim contains all of the limitations of group 7. Therefore, the arguments presented above in support of the patentability of group 7 are incorporated hereunder in support of group 8.

The group 8 claim further provides a step for implementing statistics counting to measure (i) an active time on a bus request and (ii) an execution coverage of an internal state machine. In

[29] Killian, column 2, line 35-column 3, line 2.

[30] Office Action, November 4, 2003, page 5, claim 31 arguments.

contrast, the text of Killian cited by the Examiner appears to be silent regarding statistics counting. The text of Killian cited by the Examiner reads:

> Similarly, if a TIE [Tensilica Instruction Set Extensions] coprocessor has been defined to work on different size integers, regions of the code are examined to see if all data in the region is accessed as if it were the different size.[31]

Nowhere in the above text does Killian expressly mention implementing statistics counting. Furthermore, the citation to all of FIGS. 2, and 6-8 by the Examiner is an improper omnibus argument per M.P.E.P. §707.07(d). Therefore, Killian does not disclose or suggest a step for implementing statistics counting to measure (i) an active time on a bus request and (ii) an execution coverage of an internal state machine as presently claimed. As such, group 8 is fully patentable over the cited reference and the rejection should be reversed.

Groups 1-8 are separately patentable.

During prosecution, each independent and dependent claim is considered to be separately patentable over every other claim.[32] As such, each of the above groups is considered to be separately patentable over every other group.[33] In particular, each of the groups includes a unique combination of arguments that allow individual groups to stand over the references even if all of the other groups fall.

[31] Killian, column 27, lines 32-35.

[32] See, e.g., *Rowe v. Dror*, 42 USPQ2d 1550, 1552 (Fed. Cir. 1997), *Preemption Devices, Inc. v. Minnesota Mining and Manufacturing Company*, 221 USPQ 841, 843 (Fed. Cir. 1984), and *Jones v. Hardy*, 727 F.2d 1524, 1528, 220 USPQ 1021, 1024 (Fed. Cir. 1984) (It is well established that each claim in a patent constitutes a separate invention.).

[33] M.P.E.P., Eighth Edition, Revised February 2003, §1206.

Group 2 includes an argument that Killian does not disclose or suggest a step for interfacing a chip with an FPGA as presently claimed. Since group 1 does not depend on the interfacing argument, group 2 may be found patentable even if group 1 is not.

Group 3 includes an argument that Killian does not disclose or suggest allowing a debugging/bug fix circuit observation of one or more signals when running in a normal mode as presently claimed. Since groups 1 and 2 do not depend on the observation argument, group 3 may be found patentable even if groups 1 and/or 2 are not.

Group 4 includes an argument that Killian does not disclose or suggest a debugging/bug fix circuit comprising a debugging workstation as presently claimed. Since groups 1-3 do not depend on the workstation argument, group 4 may be found patentable even if groups 1, 2 and/or 3 are not.

Group 5 includes an argument that Killian does not disclose or suggest a computer readable medium configured to store instructions as presently claimed. Since groups 1-4 do not depend on the computer readable medium argument, group 5 may be found patentable even if groups 1-3 and/or 4 are not.

Group 6 includes an argument that Killian does not disclose or suggest a step for searching for a specific signal pattern as presently claimed. Since groups 1-5 do not depend on the searching argument, group 6 may be found patentable even if groups 1-4 and/or 5 are not.

Group 7 includes an argument that Killian does not disclose or suggest a step for monitoring a correctness of a bus protocol as presently claimed. Since groups 1-6 do not depend on the monitoring argument, group 7 may be found patentable even if groups 1-5 and/or 6 are not.

Group 8 includes an argument that Killian does not disclose or suggest a step for implementing statistics counting. Since groups 1-7 do not depend on the counting argument, group 8 may be found patentable even if groups 1-6 and/or 7 are not.

B. CONCLUSION

The cited reference is not shown to expressly or inherently disclose all of the elements and steps as arranged in the claims. Hence, the Examiner has clearly erred with respect to the patentability of the claimed invention. It is respectfully requested that the Board overturn the Examiner's rejection of all pending claims, and hold that the claims are not rendered anticipated by the cited reference. However, should the Board find the arguments herein in support of independent claims 1 and/or 26 unpersuasive, the Board is respectfully requested to carefully consider the arguments set forth above in support of each of the independently patentable groups.

Respectfully submitted,

CHRISTOPHER P. MAIORANA, P.C.

Christopher P. Maiorana
Registration No. 42,829

Dated: June 3, 2004

c/o Leo Peters
Intellectual Property Law Department
LSI Logic Corporation
1621 Barber Lane
MS: D-106 Legal
Milpitas, CA 95035

IX. APPENDIX

The claims of the present application which are involved in this appeal are as follows:

1. A system for designing an integrated circuit (IC) comprising:

a functional portion;

a logic portion connected to said functional portion and configured to detect errors, fix errors or verify fixes of errors in said functional portion, wherein said logic portion includes one or more interfaces;

a debugging/bug fix circuit configured to detect errors in said logic portion through said one or more interfaces; and

a diagnostic architecture using said FPGA core in a system on a chip design.

4. The system according to claim 1, wherein said system is further configured to (i) provide ease in bringing up, (ii) verification and (iii) debugging, each by interconnecting said circuit and said debugging/bug fix circuit.

5. The system according to claim 1, wherein said system is further configured to provide one or more programming options of said circuit.

6. The system according to claim 1, wherein said system is further configured to allow observation of one or more signals by said debugging/bug fix circuit.

7. The system according to claim 6, wherein said system is further configured to allow observation of said one or more signals when running in a normal mode.

8. The system according to claim 1, wherein said system is further configured to run in a single step mode.

9. The system according to claim 8, wherein said system is further configured to run in said single step mode when controlled by a gate or a core.

11. The system according to claim 9, wherein said core is programmable.

12. The system according to claim 1, wherein said debugging/bug fix circuit comprises a debugging workstation.

13. The system according to claim 1, wherein said debugging/bug fix circuit is further configured to allow one or more debugging features.

14. The system according to claim 13, wherein said one or more debugging features support triggering and tracing based on one or more internal signals.

15. The system according to claim 13, wherein said one or more debugging features support dynamically changing host register values.

16. The system according to claim 13, wherein said one or more debugging features provide complex monitoring functions.

17. The system according to claim 1, wherein said system is further configured to reduce debugging/verification time and/or improve product time to market.

18. The system according to claim 1, wherein said circuit is further configured to operate in a normal mode and a single step mode.

19. The system according to claim 18, wherein said normal mode is configured to allow said circuit to present one or more internal signals of said functional portion and said single step mode is configured to provide a plurality of signals of said functional portion.

20. The system according to claim 18, wherein a scan chain is used to diagnose or fix a bug via the logic portion.

21. The system according to claim 19, wherein the logic portion is further configured to bridge one or more of said plurality of signals between a plurality of modules.

22. The system according to claim 2, wherein said debugging/bug fix circuit and said circuit are configured to generate one or more debugging features.

23. The system according to claim 1, wherein said debugging/bug fix circuit is configured to work with Computer Aided Design (CAD) software to provide one or more

diagnostic functions.

24. The system according to claim 23, wherein said diagnostic functions are selected from the group consisting of searching for a specific signal pattern, tracing the internal state machine, triggering on a programmed condition and other appropriate diagnostic functions.

25. The system according to claim 23, wherein said diagnostic functions are selected from the group consisting of on the fly monitoring of a correctness of a bus protocol, and implementing statistics counting to measure the performance and the testing coverage.

26. A method for diagnostics comprising the steps of:

(A) interfacing a chip with an FPGA core;

(B) presenting one or more internal signals of said chip;

(C) verifying or fixing bugs in said chip with said one or more internal signals; and

(D) programming the FPGA core to dump data from a host register every N clock cycles.

27. A computer readable medium configured to store instructions for executing the steps of claim 26.

28. The method according to claim 26, further comprising:

capturing signals every N clock cycles.

29. The method according to claim 28, further comprising the step of: dynamically changing the values in said host register.

30. The method according to claim 29, further comprising the step of: searching for a specific signal pattern.

31. The method according to claim 30, further comprising the step of: monitoring the correctness of a bus protocol.

32. The method according to claim 31, further comprising the step of: implementing statistics counting to measure (i) the active time on bus request and (ii) the execution coverage of the internal state machines.

33. The method according to claim 26, further comprising the steps of:

triggering and tracing based on internal signals;

triggering on the specific values of address, data, or command bus;

dynamically changing the host register values; and

monitoring protocol functions.